Universidade Federal do Rio de Janeiro

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

Decarlo Áureo Cerqueira de Souza

MONITORAMENTO DE REDE LOCAL: Alternativa com uso de Ferramenta Aberta Nagios

Rio de Janeiro

2014

Decarlo Áureo Cerqueira de Souza

MONITORAMENTO DE REDE LOCAL:

Alternativa com o Uso de Ferramenta Aberta Nagios

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação *Lato Sensu* em Gerência de Redes de Computadores e Tecnologia Internet do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Orientador:

Rolando Oscar Castro S, Esp., UFRJ, Brasil

Rio de Janeiro

2014

Decarlo Áureo Cerqueira de Souza

MONITORAMENTO DE REDE LOCAL:

Alternativa com o Uso de Ferramenta Aberta Nagios

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação *Lato Sensu* em Gerência de Redes de Computadores e Tecnologia Internet do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Aprovada em março de 2014.

Oscar Castro, M.Sc., UFRJ, Brasil

Dedico este trabalho *In Memoriam* aos meus pais, Vanda e Zoberto, que, na sua simplicidade, educaram os filhos com sabedoria.

AGRADECIMENTOS

Agradeço a Deus, que é o meu refúgio, permitindo-me chegar tão longe em minha caminhada.

Agradeço à CMRJ, que disponibilizou recursos e tempo, de modo a propiciar o meu aprimoramento profissional.

Agradeço especialmente a Evaldo Mello pelo apoio e companheirismo.

Agradeço também aos professores da UFRJ, que se empenharam em transmitir experiências e conhecimentos para os alunos do MOT.

Agradeço ao meu irmão Marco Polo pela colaboração e incentivo.

RESUMO

SOUZA, Decarlo Áureo Cerqueira de. **MONITORAMENTO DE REDE LOCAL: alternativa com o uso de Ferramenta Aberta Nagios.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2014.

Este trabalho discute o monitoramento de rede de computadores sob a perspectiva de padrões abertos, com vista a possibilitar uma integração dos vários níveis de atividade. Assim, descreve a evolução da arquitetura comumente conhecida como SNMP – Simple Network Management Protocol e apresenta um estudo de caso de implementação da ferramenta aberta de gerenciamento Nagios.

ABSTRACT

SOUZA, Decarlo Áureo Cerqueira de. **MONITORAMENTO DE REDE LOCAL: alternativa com o uso de Ferramenta Aberta Nagios.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2014.

The objective of this study is to discuss de network management through open standards and to enable the integration among different levels of the activity. In this way, it describes the development of an architecture commonly known as SNMP – Simple Network Management Protocol. It also presents a case study in which the open tool of management: Nagios.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

AD	Active Directory
ASN.1	Abstract Syntax Notation One
CMIP	Common Management Information Protocol
CMOT	CMIP over TCP/IP
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IAB	Internet Architecture Board
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Eletronic Engineers
IOS	Internetwork Operating System
IP	Internet Protocol
IS	Internet Standard
ISO	International Standadization Organization
ISP	Internet Service Provider
LAN	Local Area Network
MD5	Message-Digest algorithm 5
MIB	Management Information Base
MO	Managed Object
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair
MTU	Maximum Transmission Unit
NMS	Network Management Station
NOC	Network Operations Center
OID	Object Identifier
OS	Operating System
PNG	Portable Network Graphics
OSI	Open Systems Interconnection
RFC	Request for Comments
RMON	Remote Network Monitoring
SGMP	Simple Gateway Monitoring Protocol
SHA	Secure Hash Algorithm
SLA	Service Level Agreement
SMI	Structure of Management Information
SNTP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
TTL	Time to Live
TTS	Trouble Ticket System
UDP	User Datagram Protocol
PDU	Protocol Data Unit

WINS	Windows Internet Naming Service
WAN	Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

A maioria das organizações depende de um conjunto complexo de servidores e equipamentos de rede para garantir que os dados da atividade institucional possam fluir sem problemas entre os membros da instituição.

A confiabilidade da rede de computadores, a velocidade e a eficiência são cruciais para que as instituições realizem bem suas atividades, as quais são altamente dependentes dos ambientes de TI – Tecnologia da Informação.

A gerência de uma rede sem mecanismos de controle pode apresentar problemas como congestionamento de tráfego, recursos mal utilizados, recursos sobrecarregados, problemas com segurança, insatisfação dos usuários, dentre outros.

Sendo assim, a necessidade de administrar falhas, configurações, contabilizações, desempenho e segurança tem incentivado o desenvolvimento de várias ferramentas. Uma categoria que se destaca é a de ferramentas com a finalidade de monitorar e fornecer estatísticas detalhadas sobre o tráfego nesses ambientes, para que se possa, por exemplo, identificar gargalos e caracterizar o tráfego. Essas ferramentas se baseiam na análise individual dos pacotes que trafegam na rede e acabam se limitando a contabilizar o tráfego, entre outros critérios. Ao utilizar uma ferramenta de monitoramento, torna-se difícil identificar problemas relacionados ao comportamento de um protocolo de alto nível, medir o tempo de resposta de determinada transação e detectar a presença de intrusos na rede.

O gerenciador da rede visa manter o controle das informações estratégicas, controlar a complexidade, obter melhorias nos serviços, reduzir ao máximo o tempo de indisponibilidade e minimizar os custos de manutenção. Esses esforços são voltados para maximizar sua eficiência e produtividade. Porém, para que tudo isso ocorra, é necessário o uso de ferramentas de gerência e monitoramento automáticos.

O estudo deste trabalho se baseia em produzir um documento que facilite a implementação em um ambiente real, através de uso da ferramenta aberta Nagios, largamente utilizada nas redes corporativas.

1.1 MOTIVAÇÃO E OBJETIVO

Em um ambiente de rede de computadores onde os ativos estão distribuídos por dois prédios com diversos andares e salas faz-se necessário o uso de pelo menos uma ferramenta administrativa global.

A necessidade que os administradores de TI têm de saber sobre o estado dos ativos e sua localização é de suma importância para o bom funcionamento da rede, oferecendo aos seus usuários estabilidade e segurança, no que tange às informações corporativas e individuais.

O motivo de se instalar e configurar adequadamente uma ou mais ferramentas, que efetuem a análise e identifica a configuração de uma rede de computadores, é oferecer ao gestor de TI um mapa da estrutura e o comportamento de cada interface dos ativos que compõem essa rede. Desta forma, ele poderá interagir com a rede, identificando eventuais problemas e necessidades de investimentos.

O objetivo deste trabalho é implementar uma ferramenta aberta de gerenciamento de redes de computadores: o Nagios. A inexistência de ferramentas administrativas para efetuar o monitoramento global da rede dificulta a ação proativa nas eventuais interrupções da rede como um todo ou em qualquer de um dos seus segmentos.

1.2 ORGANIZAÇÃO DESTE TRABALHO

A monografia está organizada em seis capítulos.

O capítulo 1 faz uma breve introdução sobre a necessidade que as instituições têm em manter um gerenciamento de seus ativos de rede, de forma automatizada.

O capítulo 2 apresenta a gerência de redes e o modelo FCAPS (Fail, Configuration, Accounting, Performance and Security) adotado pela ISO (International Standard Organization).

O capítulo 3 trata da modelagem, análise do ambiente e as ferramentas de monitoramento de uma rede.

O capítulo 4 refere-se a um estudo de caso sobre o gerenciamento de uma rede corporativa.

O capítulo 5 descreve a implementação da ferramente de gerenciamento utilizada neste trabalho, o NAGIOS.

O capítulo 6 está reservado para as conclusões.

2 CONCEITOS BÁSICOS

Atualmente as redes corporativas são compostas de inúmeros dispositivos que necessitam estar interligados, para que haja o compartilhamento de informações e dos recursos disponíveis, de forma ágil, de tal modo que os administradores de rede saibam como elas se comportam.

Gerência de redes é o monitoramento de qualquer estrutura física e lógica de uma rede de computadores. Essa atividade é de extrema importância para que se obtenha um bom fluxo no tráfego das informações, garanta que os recursos sejam corretamente utilizados e não sobrecarregados, e que esses dados sejam transportados com confiabilidade e segurança. Tem por princípio básico detectar falhas e corrigi-las em tempo hábil, prevendo problemas futuros, sem que haja prejuízo no monitoramento.

A equipe de TI é responsável por todo o monitoramento da rede e precisa ter um bom conhecimento das especificações de *hardware* e *software* dos servidores e das estações de trabalho. Assim, essa equipe deverá conhecer e zelar pelo bom funcionamento de equipamentos, tais como: roteadores, *switches*, dispositivos móveis, controladoras de dispositivos *wireless*, dentre outros.

A gerência de redes possui quatro elementos básicos: o gerente, o software de protocolo, o agente e a base de dados. O gerente é um computador conectado à rede que executa o *software* de protocolo de gerenciamento, o qual solicita informações do agente. O agente é um *software* que roda em um recurso, elemento ou sistema gerenciado. Este exporta uma base de dados de gerenciamento (*MIB – Management Information Base*), para que o gerente tenha acesso ao mesmo.

A MIB é o conjunto de objetos gerenciados que procura abranger todas as informações necessárias para a gerência da rede, possibilitando, assim, a

automatização de grande parte das tarefas de gerência. O protocolo simples de gerenciamento de rede (*SNMP - Simple Network Management Protocol*) informa os mecanismos de comunicação entre o gerente e o agente.

2.1 O MODELO FCAPS

Para que haja êxito no funcionamento e na eficiência de uma rede corporativa, é necessário utilizar os principais componentes de um sistema de gerenciamento de redes que são: Gerência de Falhas, Gerência de Configuração, Gerência de Contabilização, Gerência de Desempenho e Gerência de Segurança.

Nesse contexto, gerenciar conforme o modelo FCAPS (Fault, Configuration, Accounting, Performance and Security) baseia-se em como encontrar formas de resolver as questões que envolvam falhas, configuração, contabilizações referentes à rede, desempenho e segurança. A figura 1 a seguir representa essas áreas do modelo FCAPS.

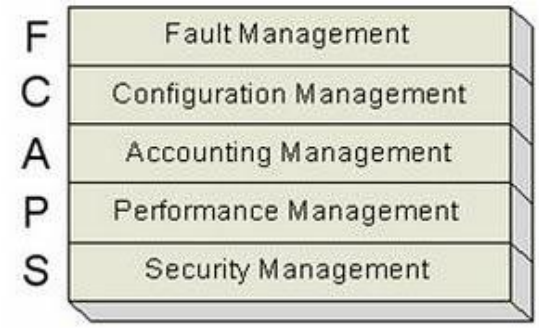

Figura 1 – Referência ao nome do modelo FCAPS

O modelo FCAPS recebe este nome por ser criado a partir das iniciais de cada área de gerenciamento:

1. Gerência de Falhas: auxilia a encontrar falhas descobertas na rede e tem por função detectar, isolar e solucionar o problema.

2. Gerência de Configuração: auxilia a encontrar a configuração de todos os dispositivos que estão relacionados à rede em questão. Busca informações sobre as configurações, geração de eventos, atribuição de valores iniciais aos parâmetros dos elementos gerenciados, registro de informações, alteração de configuração desses elementos, início e encerramento de operação dos elementos gerenciados.
3. Gerência de Contabilização: determina as formas de acesso disponibilizado aos usuários e tem como objetivo coletar informações sobre a utilização, o estabelecimento de cotas, a escala de tarifação e a aplicação de tarifas e faturamento.
4. Gerência de Desempenho: está relacionada diretamente com a qualidade de serviço da rede. Para tanto, essa gerência necessita planejar a capacidade da rede em manter e prestar suporte a todos os usuários; além disso, deve utilizar indicadores para o adequado monitoramento da rede.
5. Gerência de Segurança: estabelece o controle de todos os acessos à rede, protege os elementos e detecta possíveis tentativas de invasão.

2.2 SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

O SNMP foi criado em 1988, com a RFC 1052 (*Request for Comments*[1]), para suprir a necessidade de se padronizar o processo de gerenciamento de dispositivos

[1] RFC (Request for Comments) é um documento que descreve os padrões de cada protocolo da Internet, antes de serem considerados um padrão. Trata-se de um documento do IETF (Internet Engineering Task Force), que traz as especificações de um protocolo ou tecnologia. As RFCs são publicadas toda vez que um comitê do IETF chega a um resultado consolidado entre as partes interessadas. O IETF é uma instituição que desenvolve e promove as normas de Internet. O site do IETF é www.ietf.org e nele encontramos as especificações de diversos protocolos associados à implementação e operação da Internet. Ref.: Wikipedia.

IP (Internet Protocol), facilitando o seu gerenciamento remoto, por meio de um conjunto de regras e processos. A RFC 1052 estabeleceu os requisitos para a padronização da gerência de redes. As primeiras RFCs foram publicadas em 1998, utilizando parte dos conceitos desenvolvidos para roteadores, principalmente o SGMP (*Simple Gateway Monitoring Protocol*)

A RFC 1052 foi reescrita com novas funções e a versão 1.0 do SNMP foi publicada em meados de 1991. Muitos grupos de trabalho contribuíram para o desenvolvimento desse protocolo e criaram MIBs (*Management Information Base)* para vários tipos de equipamentos de rede. Constituem exemplos: roteadores, *bridges*, *hubs*, *switches*, monitores ASCII (*American Standard Code for Information Interchange*), interfaces WAN (*Wide Area Network*), DS1, DS3, X.25, *Frame Relay*, *Ethernet*, *Token Ring*, FDDI, etc.) e ainda protocolos proprietários.

Em resumo, pode-se considerar que “O núcleo do SNMP é um conjunto simples de operações (e das informações obtidas por essas operações) que permitam ao administrador modificar o estado de alguns dispositivos baseados em SNMP.*” (MAURO e SCHMIDT, 2001).*

2.2.1 Gerenciadores e Agentes SNMP

O protocolo SNMP possui duas entidades: o gerente e o agente. O gerente usualmente é um servidor que executa programas com o objetivo de gerenciar a rede. Também conhecido como NMS (*Network Management Station*), tem por tarefa receber as informações emitidas pelo agente. Um NMS executa as operações de *polling* e recepção dos *traps* de agentes da rede. Por sua vez, os agentes são módulos do programa de gestão hospedado no dispositivo, o qual é monitorado na rede, traduzindo e enviando as informações coletadas ao gerente.

O *polling* é toda consulta de informações de um agente, que pode ser um servidor, roteador, *switches* e outros. Todas as informações coletadas podem ser utilizadas para detecção de alguma operação errônea na rede. A *trap* é utilizada pelo agente para comunicar à NMS a ocorrência de algum problema. As *traps* são emitidas em modos assíncronos, ou seja, não se destinam a atender solicitações e consultas da NMS. (MAURO e SHMIDT, 2001).

As operações do SNMP têm por objetivo a troca de informações entre o gerente e o agente sobre os objetos gerenciados no ativo monitorado. Uma das principais características do protocolo SNMP é ser simples, pois possui um pequeno conjunto básico de operações, baseado no paradigma conhecido como “busca-armazenamento”, o que significa que as operações do protocolo SNMP são derivadas de operações básicas de busca e armazenamento.

2.2.2 O Agente SNMP

Os agentes do protocolo SNMP mostram os dados sobre a gestão dos sistemas como variáveis, tais como: a memória disponível, o nome do sistema, a roda padrão, o número dos processos correntes, etc. O sistema gerenciador deve recuperar as informações, tanto por meio de comandos definidos, como *GET, GETNEXT* e *GETBULK*, utilizados pelo agente de maneira direta, dentre outros, quanto por meio de comandos *TRAP* ou *INFORM*. As operações de configuração e de controle são utilizadas quando necessárias à infraestrutura da rede e as de monitoramento são realizadas em uma base regular.

As variáveis acessíveis via SNMP são organizadas hierarquicamente e são definidas na *Management Information Base* (*MIB*).

A figura que se segue apresenta o fluxo da informação de gestão de rede entre o agente e o gerente, os quais devem ser configurados com dois ou mais módulos de *softwares*. Um deles é o programa que habilita o transporte da informação de gerenciamento dos dispositivos e é responsável por implementar o protocolo SNMP. O outro módulo é denominado processo de gerenciamento no gerente e processo no agente.

Os serviços de processos de gerenciamento são responsáveis pelo gerenciamento dos programas na camada de aplicação e proporcionam a interface para o protocolo de gerenciamento na rede. Dessa forma, o subagente funciona como o integrador dos processos que acessam a informação requisitada pela aplicação de gerenciamento da rede e serve como interface para o protocolo.

Figura 2 – Camadas de protocolo no gerente e no agente.

2.3 A MIB

A SMI (*Structure of Management Information*) fornece a maneira de definir os objetos gerenciados e seus componentes. Um agente tem a posse de uma lista de objetos que ele controla. Um possível objeto é o *status* operacional de uma interface do roteador (*up, down* ou *testing*). Essa lista define coletivamente as informações que o gerente pode utilizar para determinar o estado do dispositivo no qual o agente reside.

Nesse contexto, pode-se pensar em um banco de dados de objetos gerenciados e rastreados pelos agentes, em que um tipo de *status* pode ser acessado pelo gerente. A esse recurso gerencial se denomina MIB.

Enquanto a SMI fornece a maneira de definir os objetos gerenciados, a MIB é uma definição que utiliza a sintaxe da SMI dos próprios objetos. Dessa forma, a MIB se comporta em relação ao objeto como um dicionário perante uma palavra: inicialmente define o nome textual de um objeto, para em seguida atribuir-lhe um significado ou mesmo uma definição.

Um agente pode implementar várias MIBs, mas todos os agentes empregam a MIB denominada MIB-II (RFC 1213). Essa norma é responsável por definir as variáveis para a estatística de determinada interface: velocidade, MTU, octetos enviados, octetos recebidos, etc., e, ainda, o *status* concernente ao sistema, como localização deste, contato, telefone, etc. Ou seja, o principal objetivo da MIB-II é fornecer a informação TCP/IP geral de gestão.

O SNMP não define as informações ou variáveis que um sistema de gestão deve oferecer, de vez que o protocolo dispõe de um *design* extensível no qual esses dados estão disponíveis, definidos pelas bases de gerenciamento (MIBs). As MIBs então descrevem a estrutura de gestão dos dados de um subsistema de um

dispositivo, por meio do uso hierárquico de denominações, as quais contêm os identificadores dos objetos (OID - *Object Identifier*). Cada OID identifica uma variável que poderá ser lida ou escrita através do SNMP, lembrando que as MIBs utilizam a notação regulamentada pela ASN.1.

A hierarquia da MIB deve ser concebida como uma árvore invertida de raiz anônima, tendo seus ramos distribuídos por diferentes organizações. A camada mais alta das OIDs da MIB é padronizada em diferentes organizações, da mesma forma que a camada mais baixa é alocada em organizações associadas. Essa hierarquia permite a gestão em todas as camadas do modelo OSI[2]. Assim como as MIBs, a hierarquia de camadas pode ser definida para cada área específica de informação e operação, sendo possível estender as aplicações em banco de dados.

Um objeto gerenciado, denominado objeto MIB, é uma das características específicas de um dispositivo gerenciado que pode assumir uma ou mais instâncias dos objetos essencialmente variáveis identificados pelas suas OIDs.

Há dois tipos de objetos gerenciados:

1. Escalares, que definem uma única instância para os objetos.
2. Tabelados, que definem várias instâncias para os objetos que são agrupados na tabela MIB.

Um exemplo de objeto gerenciado é o *atinput*, que é um objeto escalar e seu valor inteiro indica o número total de pacotes de entrada na interface do roteador.

[2] O Modelo OSI - Open Systems Interconnection, criado em 1995 e formalizado em 1983, é um modelo de referência da ISO - International Organization for Standardization, cujo principal objetivo era ser um modelo *standard*, para protocolos de comunicação entre os mais diversos sistemas, e assim garantir a comunicação *end-to-end*. A arquitetura OSI divide as redes de computadores em 7 camadas, de forma a se obter camadas de abstração. Cada protocolo implementa uma funcionalidade assinalada a uma determinada camada. O Modelo OSI permite comunicação entre máquinas heterogêneas e define diretivas genéricas para a construção de redes de computadores, seja de curta, média ou longa distância, independente da tecnologia utilizada. Ref.: Wikipedia.

Um OID identifica unicamente um objeto gerenciado na hierarquia da MIB. Ressalte-se que existem MIBs proprietárias desenvolvidas pelos fabricantes dos dispositivos e muitas propostas e projetos têm sido desenvolvidos para ajudar o gerenciamento, tais como: *frame relay, ATM, FDDI* e outros.

A seguir são apresentados alguns conceitos de objetos gerenciados.

Um objeto gerenciado é uma visão abstrata de um recurso real do sistema e, dessa forma, todos os recursos da rede que devem ser gerenciados são modelados. As estruturas dos dados resultantes dessa modelagem são os objetos gerenciados. Esses objetos poderão ter permissões de serem lidos e alterados, sendo que cada leitura representa um estado real do recurso, cuja alteração refletirá o estado daquele recurso.

A MIB é um banco de dados de todos os dispositivos gerenciados que o agente monitora, cujo objetivo é definir um nome e um texto de um dos objetos gerenciados, e trazer a explicação de seu significado. A MIB-I é a original, mas não é mais consultada desde a implementação da MIB-II, que passou a ser utilizada.

Os agentes podem instalar inúmeras MIBs, porém é necessário implementar uma MIB específica, a MIB-II. Nesta são encontrados todos os dados estatísticos necessários, tais como a unidade fundamental de transferência de rede TCP/IP, a velocidade da interface e a MTU (*Maximum Transmission Unit*), dentre outros. A MIB-II visa apresentar todas as informações sobre o gerenciamento de TCP/IP.

O nó da raiz hierárquica da MIB não possui rótulo e possui pelo menos três subníveis, que são: o nó 0, administrado pela CCITT (*Consultative Committe for International Telegraph and Telephone*); o nó 1, administrado pela ISO (*International Standard of Organization*); e o nó 2, administrado em conjunto pela CCITT e a ISO. Sob a ISO fica o nó que pode ser utilizado por outras instituições: o *org* (3), sob o qual

fica o *dod* (6), pertencente ao Departamento de Defesa dos EUA, sendo que este alocou um sub-nó para a comunidade *internet*, a qual é administrado pela IAB (*International Activities Board*). Abaixo deste nó, dentre outros, encontram-se o *management,* o *experimental* e o *private*.

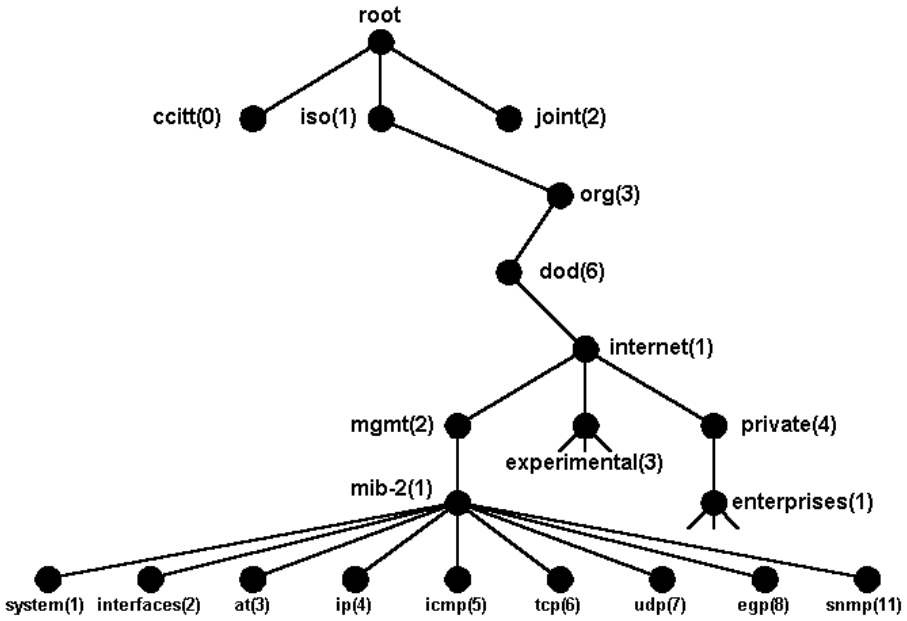

Figura 3 – Estrutura da MIB II.

Sob o nó *management* ficam as informações de gerenciamento e é sob este que se situa o nó da MIB II. Debaixo do nó *experimental* encontram-se as MIBs experimentais. Por sua vez, o nó *private* abrange o nó *enterprises*, o qual compreende os nós da indústria de equipamentos.

A seguir, apresenta-se um exemplo de um OID, o ipInReceives do grupo IP, com seus respectivos atributos identificadores:

ipInReceives Object Type

Object Identifier: 1.3.6.1.2.4.3

Access: read-only

Syntax: Counter32

Description: O número total de *datagramas* que chegam às interfaces, incluindo aqueles com erro.

2.3.1 MIB II

A subárvore MIB II organiza os objetos usados para obter informações específicas dos dispositivos da rede. Esses objetos estão divididos em 10 grupos, conforme indicado no quadro abaixo:

Quadro 1 – Os grupos de objetos da MIB II

Grupo	**Informação**
System (1)	Informações básicas do sistema
Interfaces (2)	Interfaces de rede
AT (3)	Tradução de endereços
IP (4)	Protocolo IP
ICMP (5)	Protocolo ICMP
TCP (6)	Protocolo TCP
UDP (7)	Protocolo UDP
EGP (8)	Protocolo EGP
Transmission (10)	Meios de transmissão
SNMP (11)	Protocolo SNMP

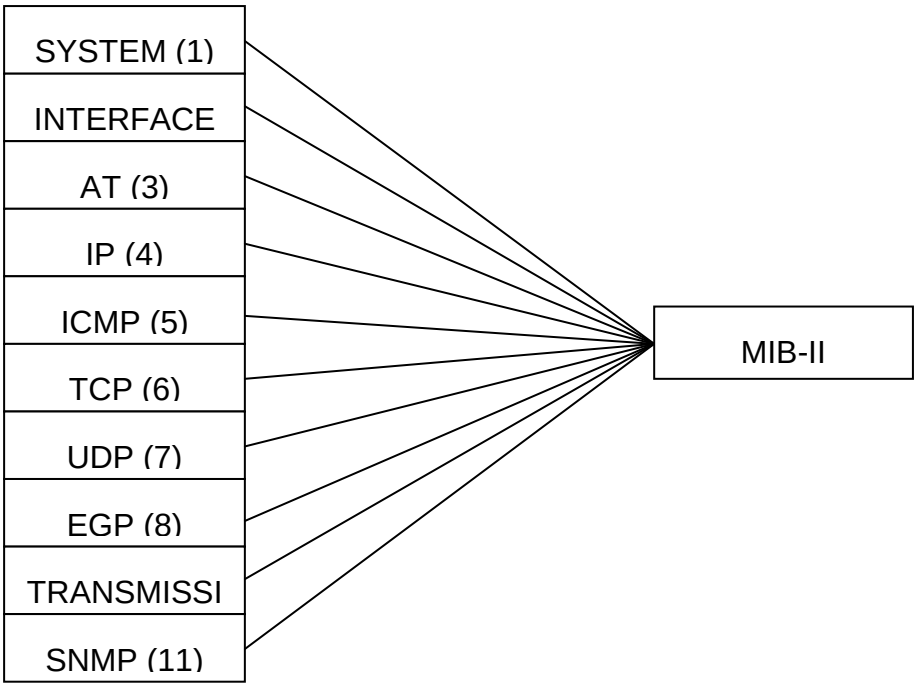

Figura 4 – MIB II

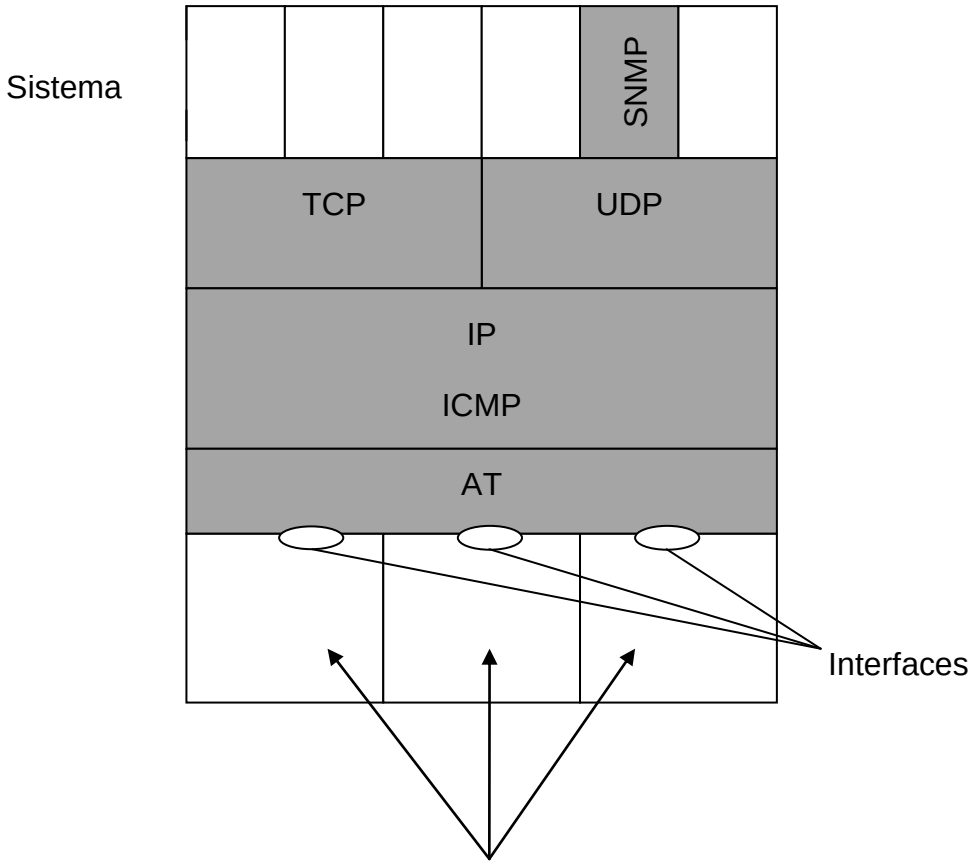

Figura 5 - Planificação do nó da MIB II

Em seguida, são indicados alguns dos principais objetos dos nós pertencentes aos grupos da MIB II:

Grupo System (1.3.6.1.2.1.1)

1. sysDescr (1.3.6.1.2.1.1.1): Descrição textual da unidade. Esse atributo do objeto pode incluir o nome e a versão do *hardware*, sistema operacional e o programa de rede.
2. sysUpTime (1.3.6.1.2.1.1.3): Tempo decorrido, em milhares de segundos, desde a última re-inicialização do gerenciamento do sistema na rede.
3. sysContact (1.3.6.1.2.1.1.4): Texto de identificação do gerente da máquina gerenciada e forma de contato com o mesmo.

Grupo Interfaces (1.3.6.1.2.1.2)

1. ifNumber (1.3.6.1.2.1.2.1): Número de interfaces de rede, não importando seu atual estado, presentes no sistema.
2. ifOperStatus (1.3.6.1.2.1.2.2.1.8): Estado atual da interface.
3. ifInOctets (1.3.6.1.2.1.2.2.1.10): Número total de octetos recebidos pela interface.

Grupo IP (1.3.6.1.2.1.4)

1. ipForwarding (1.3.6.1.2.1.4.1): Indica se esta entidade é um *gateway*.
2. ipInHdrErrors (1.3.6.1.2.1.4.4): Número de datagramas que foram recebidos e descartados devido a erros no cabeçalho IP.

Grupo ICMP (1.3.6.1.2.1.5)

1. icmpInMsgs (1.3.6.1.2.1.5.1): Número total de mensagens ICMP recebidas por esta entidade.
2. icmpOutMsgs (1.3.6.1.2.1.5.14): Número total de mensagens ICMP enviadas por esta entidade; incluindo aquelas com erros.

Grupo TCP (1.3.6.1.2.1.6)

1. tcpMaxConn (1.3.6.1.2.1.6.4): Número máximo de conexões TCP que esta entidade pode suportar.
2. tcpCurrentEstab (1.3.6.1.2.1.6.9): Número de conexões TCP que estão como estabelecidas ou a espera de fechamento.
3. tcpRetransSegs (1.3.6.1.2.1.6.12): Número total de segmentos retransmitidos.

Grupo UDP (1.3.6.1.2.1.70

1. udpInDatagrams (1.3.6.1.2.1.7.1): Número total de datagramas UDP entregues aos usuários UDP.
2. udpNoPorts (1.3.6.1.2.1.7.2): Número total de datagramas recebidos para os quais não existia aplicação na referida porta.
3. udpLocalPort (1.3.6.1.2.1.7.5.1.2): Número da porta do usuário UDP local.

Grupo SNMP (2.3.6.1.2.1.11)

1. snmpInPkts (1.3.6.1.2.1.11.1): Número total de mensagens recebidas pela entidade SNMP.

2. snmpOutPkts (1.3.6.1.2.1.11.2): Número total de mensagens envidadas pela entidade SNMP.
3. snmpInTotalReqVars (1.3.6.1.2.1.11.13): Número total de objetos da MIB que foram resgatados pela entidade SNMP.

2.4 SNMPv1

O protocolo de comunicação de interfaces SNMPv1 SMI especifica o uso de uma série de tipos de dados SMI-*specific*, os quais são divididos em duas categorias:

1) Tipos de dados simples:

- Três tipos de dados simples são definidos na SMI SNMPv1, cujos valores são invariáveis:
  a) *Interger* - Dado inteiro no intervalo de -2^{31} a 2^{31} -1.
  b) *Octer strings* - Sequências ordenadas de 0 a 65.535 octetos.
  c) *Object IDs* - Conjunto de todos os identificadores de objetos, atribuídos de acordo com as regras especificadas no ASN.1.

2) Tipos de dados de aplicação vasta:

- Sete tipos de dados de aplicação vasta existem no SNMPv1 SMI: endereços de rede, contadores, medidores, *time ticks*, opacos, inteiros (*interger*) e inteiros sem sinal.
  a) Os endereços de rede representam endereços de uma família particular de protocolos. O SNMPv1 suporta apenas os endereços IP de 32 bits.
  b) Os contadores são inteiros não-negativos que aumentam até atingir um valor máximo e depois retornam a zero. Em SNMPv1, um tamanho de 32 bits do contador é especificado.

c) Os medidores são inteiros não-negativos que podem aumentar ou diminuir entre valores mínimos e máximos. Sempre que a propriedade do sistema apresentada pelo indicador está fora da faixa, o valor do indicador não varia mais que seu máximo nem menos que seu mínimo, conforme especificado na RFC 2578.

d) Um *time tick* é um centésimo de segundo desde a ocorrência de um evento.

e) Os opacos representam uma codificação arbitrária usada para passar *strings* de informações arbitrárias que não estejam estritamente em conformidade com a linguagem usada pelo SMI.

f) Um *interger* representa um inteiro assinado com valor de informação. Esse tipo de dado redefine o tipo de dado *interger*, que tem a precisão arbitrária no ASN.1, porém tem precisão limitada no SMI.

g) Um *interger unsigned* representa valores de inteiros de informação em módulo e é útil quando os valores são sempre não-negativos. Esse tipo de dado redefine o tipo de dado *interger*, que tem sua precisão limitada no SMI, porém arbitrária no ASN.1.

O SNMPv1 SMI define quadros altamente estruturados que são usados para agrupar as instâncias de um objeto tabelado, ou seja, um objeto que contém múltiplas variáveis. As tabelas são compostas de zeros ou mais linhas, que são indexadas de modo a permitir ao SNMP a recuperação ou a alteração de uma linha inteira, por meio de um único comando, o qual pode ser: *Get, GetNext* ou *Set.*

2.4.1 SNMPv2

O SMI SNMPv2 está descrito na RFC 2578. Esse protocolo realiza algumas adições e melhorias em relação à versão anterior, o SNMPv1 SMI-*especific,* tais como: inclusão de *bit strings*, endereços de rede e contadores. As *bit strings* são definidas apenas em SNMPv2 e englobam zeros ou mais *bits* nomeados que especificam um valor. No SNMPv1, os contadores têm um tamanho fixo de 32 *bits*, porém, no SNMPv2, os contadores podem ser definidos também com o tamanho de 64 *bits*.

O protocolo SNMP opera na camada de aplicação (*layer* 7) do modelo OSI. O protocolo especifica, na sua primeira versão, cinco unidades básicas de dados, as PDUs, conforme a seguir indicado:

1. *GET REQUEST* – usado para recuperar um pedaço de informação de gestão.
2. *GETNEXT REQUEST* – usado iterativamente para recuperar sequências de gestão da informação.
3. *GET RESPONSE* – usado pelo agente para responder sobre que dados deve obter e definir os pedidos pelo gestor.
4. *SET RESPONSE* – usado para iniciar e fazer uma mudança para um valor do elemento da rede.
5. *TRAP* – utilizado para relatar um caso de alerta ou outros eventos assíncronos sobre um subsistema. No SNMPv1, relatórios de eventos assíncronos são chamados de *traps*, ao passo que nas versões posteriores do SNMP são denominados *notificações*. Nos módulos MIB do SNMPv1, as *traps* são definidas usando-se a macro *TRAP-TYPE*; em módulos MIB do

SNMPv2, essas *traps* são definidas usando-se a macro *NOTIFICATION-TYPE*.

Outras *PDUs* foram adicionadas na versão SNMPv2, que são:

1. *GETBULK REQUEST* – um rápido iterador usado para recuperar sequências de informação da gestão.
2. *INFORM* – reconhecimento de uma trap.

O quadro que se segue apresenta as operações do SNMPv2:

Quadro 2 – As operações do SNMPv2

Operação	**Interação**	**Descrição**
GetRequest GetNextRequest GetBulkRequest	Gerente – Agente	Solicita a leitura sobre o conteúdo dos objetos: • Apenas uma instância de objeto • O próximo da lita • O bloco inteiro
InformRequest	Gerente – Agente	Indica o valor da MIB disponível a partir da versão 2 do SNMP.
SetRequest	Gerente – Agente	Define o valor da MIB.
Response	Agente – Gerente	Retorna o conteúdo do valor em resposta ao pedido do gerente.
Trap	Agente – Gerente	Informa o gerente a ocorrência de um evento excepcional.

Normalmente o SNMP utiliza as portas 161 para o agente e 162 para o gerente. O gestor pode enviar solicitações de qualquer porta de origem disponível para a porta 161 no agente (porta de destino). A resposta do agente será reportada de volta para a porta de origem. O gestor receberá *traps* na porta 162. Deve-se destacar que o agente pode gerar as *traps* de qualquer porta disponível.

O SMI no SNMPv2 também especifica módulos de informação, os quais organizam um conjunto de definições relacionadas. Existem três tipos de módulos de informação no SMI: módulos MIB, declarações de conformidade e as declarações de capacidade.

Os módulos MIB contêm definições de objetos gerenciados relacionados entre si. As declarações de capacidade são usadas para indicar o nível preciso e apoio que um agente reivindica no que diz respeito a um grupo de MIB. A NMS pode ajustar seu comportamento em relação aos agentes, de acordo com as declarações de capacidade associada a cada agente.

Os protocolos SNMPv1 e SNMPv2 utilizam a noção de comunidades para estabelecer um grau de confiança entre os agentes e os gerentes.

Um agente é configurado com três nomes na comunidade: *read-only* (somente leitura), *read-write* (leitura/escrita) e *trap.* Os nomes dessas comunidades são essencialmente senhas. Não há diferença real entre um caractere da comunidade e a senha usada para acessar uma conta no computador.

Os três nomes de comunidade controlam diferentes tipos de atividades. Como o próprio nome indica, a sequência de comunidade *read-only* permite ler valores de dados, mas não permite que o usuário os modifique. Por exemplo, o agente permite que o gerente leia o número de pacotes que tenham sido transferidos através das portas em um roteador, mas não permite alterar os valores dos contadores. O nome *read-write* da comunidade tem permissão para ler e modificar os valores de dados. Por meio desse recurso o gerente pode ler os contadores, redefinir seus valores e até redefinir as interfaces, ou empreender outras ações capazes de mudar a configuração do roteador. Em última análise, o *trap* permite que o gerente receba notificações assíncronas do agente.

2.4.2 SNMPv3

A segurança tem sido a maior fraqueza do SNMP desde o início. A autenticação nas versões 1 e 2 equivale a uma simples senha, *string* de comunidade, enviada em texto não criptografado entre o gerente e o agente. Todo administrador de um sistema de rede de computadores, dotado com um grau de segurança aceitável, tem consciência de que senhas de texto simples não oferecem uma segurança eficaz. Não é difícil para alguém capacitado e mal-intencionado interceptar a *string* de comunidade. Uma vez que isto aconteça, a *string* poderá ser usada para se obter informações de dispositivos da rede, modificar suas configurações e, até mesmo, fechá-los.

O SNMPv3 aborda os problemas de segurança que têm atormentado tanto o SNMPv1 e o SNMPv2. Para todos os efeitos práticos, os endereços do SNMPv3 são as únicas mudanças orientadas para a segurança e não existem outras alterações no protocolo e nem novas operações. O SNMPv3 suporta todas as operações definidas pelas versões 1 e 2, porém, há novas convenções textuais, mas estas são realmente apenas formas mais precisas de interpretar os tipos de dados que foram definidos em versões anteriores.

A *engine* é um recurso do SNMPv3 composto de quatro subsistemas, a saber: o expedidor, o de processamento de mensagem, o de segurança e o de controle de acesso. O trabalho do distribuidor é enviar e receber mensagens; ele tenta determinar a versão de cada mensagem recebida, isto é, v1, v2 ou v3, e se a versão é suportada pelo dispositivo OID. Esse subsistema envia a mensagem de fora para o subsistema de processamento de mensagens. O expedidor também envia mensagens SNMP a outras entidades.

O subsistema de processamento prepara as mensagens a serem transmitidas a fim de enviá-las adequadamente e extrai dados das mensagens recebidas. Esse subsistema pode conter vários módulos de processamento de mensagens. Por exemplo, um subsistema pode ter os módulos de processamento SNMPv1, SNMPv2 e SNMPv3 pedidos; também pode contar com um módulo de processamento de outros modelos que ainda não foram definidos.

O subsistema de segurança fornece serviços de autenticação e privacidade. A autenticação usa *strings* ou comunidade SNMPv1 e SNMPv2, ou ainda, autenticação baseada no usuário SNMPv3. A autenticação baseada em usuários utiliza o MD5 ou o SHA, algoritmos criados para criptografar as senhas dos usuários. O serviço de privacidade usa o algoritmo DES para criptografar e decriptografar as mensagens SNMP. Atualmente, o DES é o algoritmo utilizado, mas outros podem ser adicionados futuramente.

O subsistema de controle de acesso é responsável por controlar o acesso a objetos MIB. O gerente pode controlar quais objetos um usuário pode acessar e as operações que ele está autorizado a realizar sobre os objetos. Por exemplo, ele pode limitar um usuário de *read-write* o acesso a certas partes da árvore MIB-2, permitindo o acesso somente leitura para toda árvore.

O SNMPv3 é definido pelo RFC 3411 e a RFC 3418, conhecida como STD0062. O SNMPv3 adicionou principalmente mais segurança e aperfeiçoamentos na configuração remota em relação à versão anterior desse protocolo. O SNMPv3 é a versão atual e padrão do SNMP a partir de 2004.

O IETF tem designado para o SNMPv2 um completo *Internet Standard (IS)*, o maior nível de maturidade de uma RFC. O *IS* considera as versões anteriores

obsoletas. Em dezembro de 1997, o “Simple Times” publicou uma série de artigos escritos pelos editores RFC do SNMPv3, explicando algumas idéias subjacentes às especificações da versão 3 do protocolo.

O SNMPv3 fornece funcionalidades de segurança importantes:

1. Garantia da integridade da mensagem, ou seja, garantia de que um pacote não foi alterado em trânsito.
2. Autenticação para verificar se a mensagem é de uma fonte válida.
3. Criptografia de pacotes para evitar espionagem por uma fonte não autorizada.

Exemplo de algumas aplicações que usam o SNMP:

1. Acompanhamento em tempo real do dispositivo (*sysUpTimeInstance*).
2. Inventário de versões de OS (*sysDescr).*
3. Recolhimento de informação na interface (*ifName, ifDescr, ifSpeed, ifType, ifPhysAddr*).
4. Medição de transferência de interface de rede (*ifInOctets, ifOutOctets*).
5. Consulta de um cache ARP remoto (*ipNetToMedia*).

O primeiro objetivo da análise de segurança é o controle do acesso a alguns recursos da rede e aos hospedeiros. Outro objetivo é a prevenção de ataques que podem comprometer a rede e auxiliar a detecção de invasores, posto que os hospedeiros podem sofrer negação de serviço (*denial of service*) ou permitir que

hackers tenham acesso a sistemas como folha de pagamento, financeiro e dados de código-fonte.

3 FERRAMENTAS DE MONITORAMENTO

1) SolarWinds

O SolarWinds é composto de várias ferramentas que monitoram o tráfego em *roteadores* com controle de *banda*. Trata-se de um sistema capaz de gerar um inventário da rede, identificando seus dispositivos por IP, e dotado de capacidade de detectar quedas nos servidores. A ferramenta tem custo elevado.

Site oficial: www.solarwinds.com.br

2) WhatsUP Gold

É uma ferramenta de monitoramento de rede de baixo custo voltada para a verificação e detecção de falhas. O sistema pode garantir a integridade e a estabilidade da rede, dos *links* de comunicação e de *serviços* críticos.

Site oficial: www.whatsupgold.com

3) Nagios

Atualmente é a ferramenta de monitoramento mais conhecida por ser *open source* (código-fonte aberto) e distribuída sob a licença GPL(*General Public License*), a qual é designada para *software* livre como o *Linux*. Foi idealizada por Richard M.Stallman no projeto GNU, da FSF(*Free Software Foundation*).

O Nagios permite a integração com *plugins* que podem ser desenvolvidos pelos usuários e facilitar o monitoramento da rede. Os *plugins* podem ser desenvolvidos em *Bash*, *Perl*, *Python*, *PHP* e outras linguagens de programação.

Com o Nagios é possível instalar vários *front-end*, que são também *open souce*, os quais são desenvolvidos pela *Nagios Community*. Os *front-end* não são

requeridos na instalação do Nagios, mas são boas opções de administração da rede, como o *NagiosQL* ou o *Centreon*.

Site oficial: www.nagios.org

4) Zabbix

O *Zabbix* é uma ferramenta que tem evoluído muito nos últimos anos. Tem como característica a capacidade de monitorar diversos parâmetros de uma rede como a integridade e o desempenho dos servidores. O *Zabbix* oferece ótimos relatórios para a visualização dos dados de recursos com base nas informações recolhidas.

O *Zabbix* tem como características:

a) Interface de gerenciamento *Web* de fácil utilização.

b) *Software Open Source* distribuído pela licença GPLv2.

c) Suporte a *Linux, Solaris, HP-UX, AIX, FreeBSB, OpenBSB, MacOS X, etc.*

d) Integração com banco de dados *MySQL*, *Oracle, PostgreSQL* ou *SQLite*.

e) Suporte nativo ao protocolo *SNMP.I*

f) Monitoramento distribuído com administração Web centralizada.

g) Envio de alertas para *e-mail*, *SMS*, ou *Jabber*.

h) Controle de acesso e permissões dos usuários.

Site oficial: www.zabbix.com

5) Zenoss

O *Zenoss* tem como principais características:

a) Interface de gerenciamento *Web* de fácil utilização.

b) *Software Open Source* distribuído pela Licença GPLv2.

c) Suporte a *Linux, Mac OS X, Windows*, dentre outros.

d) Integração com banco de dados *MySQL* e *RRDtools*.

e) Não possui *agente* próprio.

f) Monitora serviços (*HTTP, POP3, IMAP*) com o uso do *SNMP, SSH, TELNET* e *WMI*.

g) Envio de alertas para *e-mail* e *SMS*.

h) Controle de acesso e permissões para os usuários.

i) Integração com o *Google Maps*®.

j) Monitoramento distribuído com os *ZenPacks*.

Site oficial: community.zenoss.org

6) PandoraFMS

Principais características do *Pandora*.

a) Interface de gerenciamento *Web* de fácil utilização.

b) *Software Open Source* distribuído pela Licença GPLv2.

c) Suporte a *Linux, Solaris, HP-UX, AIX, BSB, IPSO, OpenWRT, Windows,* etc.

d) Agente disponível para *Linux, Solaris, HP-UX, BSB, IPOS, OpenWRT, Windows 2000/XP/2003/Vista*.

e) Integração com banco de dados *MySQL, Oracle, PostgreeSQL* ou *SQLite.*

f) Monitora serviços (*HTTP, POP3, IMAP, SSH*) sem o uso de *agentes*.

g) Suporte nativo ao protocolo *SNMP.*

h) Monitoramento distribuído com administração *Web* centralizada.

i) Envio de alertas para *e-mail* e *SMS*.

j) Controle de acesso e permissões dos usuários.

4 ESTUDO DE CASO

4.1 CENÁRIO

O cenário definido para o estudo de caso deste trabalho considerou uma instituição em que a rede de dados computacionais é composta por cerca de 1.200 nós, distribuídos em três prédios, A, B e C, onde cada prédio representa um segmento. Os prédios B e C são ligados ao prédio A através de *links* Gigabit Ethernet.

4.2 A INFRAESTRUTURA

A seguir, descreve-se a distribuição dos dispositivos da rede corporativa, dos servidores e das aplicações disponíveis.

Prédio A: 11 andares, 2 *switches* Cabletron ELS-10 27x em cada andar, cada um com uma porta EPIM FX-100 (fibra ótica multimodo) de *uplink* para o equipamento central da rede, localizado no último andar. Um *switch* HP v1910 em cada andar, ligado em cascata com um dos *switches* Cabletron ELS-10. O equipamento central: Cabletron Smart Switch Router SSR-8. Em um *slot* FastEthernet deste equipamento estão ligados todos os computadores-servidores da instituição.

Prédio B: 10 andares, 1 *switch* Cabletron ELS-1027x em cada andar, cada um com uma porta de *uplink* FastEthernet para o *switch* central do prédio, um equipamento Cabletron Smart Switch Router SSR-2 (nível 3), localizado no penúltimo andar.

Prédio C: Este prédio possui um *switch* central SSR-2 Cabletron Switch Router (nível 3), ao qual estão ligados 5 *bridges wireless* Roam About R2, da Enterasys.

4.3 INTERCONEXÕES

Existem duas saídas para Internet pelo prédio A: uma de 2 Mbps como enlace redundante via Rede Rio, através de um *firewall* Cisco ASA5520 e um roteador Cabletron Smart Switch Router SSR-2; e outra de 40 Mbps, através de um *firewall*, também Cisco ASA5520, e um roteador Cisco via serviço de operadora ISP.

4.4 APLICAÇÕES

Na infraestrutura acima descrita são disponibilizados os seguintes serviços: correio eletrônico, internet, servidores de arquivos e aplicativos, aplicações Notes, intranet, *streaming* de vídeo, acesso a bases de dados MS-SQL Server 2000 e MS-SQL Server 2008, Lotus Notes e PostgreSQL.

Além dos serviços mencionados acima, os quais são orientados aos usuários finais, existem aqueles que formam o núcleo da rede que garante a interconectividade e a operação do *backbone* e seus elementos componentes: *DHCP*, *AD*, *Kerberos*, *SMTP*, *DNS* e *WINS.*

Todos os equipamentos de rede são gerenciáveis, isto é, possuem agentes de SNMPv1 e SNMPv2 nativos. Os servidores de plataforma Windows também são gerenciáveis por SNMPv1, através do agente próprio do Windows ou do *daemon* Net-SNMP, suportando também as versões 2 e 3.

Os únicos mecanismos de monitoramento utilizados até o advento do SNMP eram feitos através do programa *freeware* FreePing, que testa a conectividade de determinados *hosts*, e de uma ferramenta própria do *framework* Lotus Notes (Notes Administrator), a qual monitora alguns parâmetros de tarefas Notes como HTTP, SMTP e replicações de bases. A primeira ferramenta, entretanto, não executa verificações de serviços, mas apenas a conectividade dos *hosts*, baseando-se no

simples envio e recebimento de pacotes ICMP. Todavia, o Notes Administrator possui a limitação de verificar somente a execução de tarefas de servidores Domino.

4.5 REQUISITOS DE MONITORAMENTO

O ambiente heterogêneo descrito acima encontra-se no limiar entre estruturas de pequeno e médio portes. Nesse contexto, certo grau de complexidade apresenta-se para o seu gerenciamento adequado. O grande número de serviços ofertados e o número de usuários, aproximadamente 1.500, faz com que elevados tempos de indisponibilidade dos recursos causem transtornos à instituição. Um tempo reduzido de recuperação está associado, sem dúvida, ao acompanhamento constante de determinados parâmetros de operação de serviços e dispositivos capazes de indicar e localizar a ocorrência de falhas.

O principal objetivo deste estudo de caso é a implementação de uma ferramenta que automatize a análise de informações para detecção de falhas. Além disso, pretende-se alcançar outros objetivos, não menos importantes, conforme se relaciona a seguir:

a) compreender a hierarquia da rede e a inter-relação de seus serviços e equipamentos, a fim de proporcionar a localização precisa de eventuais problemas;
b) gerar mapas que reflitam a hierarquia da topologia e facilitem a identificação visual de objetos em situação crítica nessa hierarquia;
c) proporcionar maior flexibilidade quanto às formas possíveis para notificação aos administradores da rede;
d) propiciar um conhecimento mais preciso do comportamento do sistema de rede com o registro de atividades e mudanças de comportamento;

e) inserir mecanismos operacionais capazes de agir de forma proativa na resolução de problemas;

f) trabalhar com padrões abertos de *softwares*, de modo a propiciar um alto grau de adequação do sistema aos mais diversos ambientes (customização) e integração com outras ferramentas;

g) possibilitar a análise de performance de serviços e dispositivos;

h) favorecer o acompanhamento dos problemas por diversos operadores; e

i) produzir estatísticas e mapas de indisponibilidades de objetos gerenciados, visualizando atividades de manutenção de serviços e *hosts*.

5 IMPLANTAÇÃO

Para a implantação do sistema de monitoramento da rede no presente estudo de caso, foram escolhidos o sistema operacional Ubuntu Server 13.10, e a ferramenta Nagios.

O criador do Nagios[3], Ethan Galstad, acredita que uma das grandes vantagens dessa ferramenta reside no fato da mesma existir há muito tempo e estar continuamente em processo de aperfeiçoamento. Os técnicos do setor de TI sabem de sua existência e que se trata de uma tecnologia confiável. O Nagios tem centenas de extensões e complementos *open source* e gratuitos, todos de larga utilização e bom desempenho. Esses aspectos tornam a ferramenta atrativa para ser empregada em redes corporativas de pequeno, médio e grande portes.

Atualmente existe uma rede de desenvolvedores e de suporte internacional ao Nagios. No Brasil, por exemplo, existem centenas ou milhares de pessoas que utilizam o programa e podem contar com o suporte em língua portuguesa para dirimir dúvidas sobre o funcionamento do aplicativo.

Em última análise, as vantagens do Nagios resumem-se à possibilidade de assistência no próprio idioma em que o sistema é instalado, à gratuidade de obtenção de códigos-fonte e extensões, e à confiabilidade e alto desempenho proporcionados. Evidentemente, existem produtos similares no mercado, até mesmo de código aberto. Entretanto, na opinião do Autor deste trabalho, esses produtos, mesmo aqueles comercializados, ainda não atingiram o grau de excelência do Nagios, razão pela qual essa ferramenta foi escolhida para aplicação neste estudo de caso.

---

[3] http://www.linuxnewmedia.com.br/lm/entrevista/o_futuro_do_nagios

5.1 A CONFIGURAÇÃO DO HARDWARE

A instalação do sistema operacional será efetuada em uma máquina virtual Ubuntu 13.10 embarcado no Oracle VM Virtual Box Manager com as especificações abaixo:

Micro computador Pentium Dual-Core E5500 2.80 GHz

Memória RAM de 1.5 Gb

Controladora SATA de 16 Gb

5.2 O SISTEMA OPERACIONAL

Através do comando: *cat /etc/*-release*, obtém-se a especificação da distribuição do OS (*Operating System*) instalado no computador, que são:

DISTRIB_ID=Ubuntu

DISTRIB_RELEASE=13.10

DISTRIB_CODENAME=saucy

DISTRIB_DESCRIPTION=”Ubuntu 13.10”

NAME=”Ubuntu”

VERSION=”13.10, Saucy Salamander”

ID=ubuntu

ID_LIKE=debian

Com o comando: “uname –a” e “uname –mrs” obtém-se a versão do *kernel* do OS.

Linux ubuntu 3.11.0-12-generic

5.3 O NAGIOS

O Nagios oferece o recurso de monitoramento de sistemas de redes e requer, para sua plena utilização, a instalação de vários *plug-ins* disponíveis no *site* www.nagios.org.

O Nagios foi desenvolvido para rodar sobre a plataforma Linux, porém, existem versões específicas para as distribuições *Debian*, *SUSE*, *Fedora* e *Ubuntu*. A

ferramenta possui duas versões: uma comercial com recursos gráficos (*dashboards*), configurações efetuadas via web, reportes avançados e visualização dos dados coletados; e a versão *free* (*Open Source*), que contém somente o mecanismo de monitoramento via CLI (*Command Line Interface*), alertas básicos e alguns relatórios.

Com o Nagios é possível monitorar temperatura, iluminação e corrente elétrica nos dispositivos. Para cumprir essa tarefa, foi desenvolvido um sensor que coleta essas informações; a ferramenta captura os dados desse sensor e apresenta um gráfico das condições do ambiente.

O Nagios tem a capacidade de monitorar os recursos de clientes tais como: utilização da CPU, uso do disco, utilização de memória e outros. Com *plug-ins* específicos, pode-se monitorar serviços como SMTP, HTTP, PING, SSH, NNTP, etc. Outro recurso da ferramenta é a possibilidade de se construir uma hierarquia de rede de clientes usando clientes pais aos quais os clientes filhos estão relacionados.

5.3.1 A instalação do Nagios

A fim de facilitar o procedimento e evitar a compilação manual dos programas, escolheu-se fazer a instalação através do pacote automatizado:

 # apt-get install nagios3

Depois de efetuado tal procedimento, verifica-se que os pacotes são instalados em /etc/nagios3/, onde se encontram os arquivos apache2.conf, cgi.cfg, commands.cfg, htpasswd.users, nagios.cfg e resource.cfg.

A pasta conf.d contém os seguintes arquivos de configuração: contacts_nagios2.cfg, extinfo_nagios2.cfg, generic-host_nagios2.cfg, generic-service_nagios2.cfg, hostgoups_nagios2.cfg, localhost_nagios2.cfg sercices_nagios2.cfg e timeperiods_nagios2.cfg.

A pasta stylesheets contém os arquivos css (*cascading style sheets*), que compõem a estrutura de *layout* do Nagios.

A seguir são apresentados os arquivos de configuração do Nagios: contacts_nagios2.cfg, extinfo_nagios2.cfg, generic-host_nagios2.cfg, generic-service_nagios2.cfg, hostgroups_nagios2.cfg, localhost_nagios2.cfg, services_nagios.cfg e timeperiods_nagios2.cfg.

A configuração do arquivo **contacts_nagios2.cfg** contém os contatos que serão notificados pelo Nagios.

```
Define contact{
        contact_name                    root
        alias                           Root
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,r
        service_notification_commands   notify-service-by-email
        host_notification_commands      hotify-host-by-email
        email                           root@localhost
        }
# Contact Groups

Define contactgroup{
        contactgroup_name               admins
        alias                           Nagios Administrators
        members                         root
        }
```

A configuração do arquivo **extinfo_nagios2.cfg** contém configurações avançadas de *hosts* e *serviços*:

```
define hostexinfo{
        hostgroup_name                  debian-servers
        notes                           Debian GNU/Linux servers
#       notes_url
        http://webserver.localhost.localdomain/hostinfo.pl?host=netware1
        icon_image                      base/debian.png
        icon_mage_alt                   Debian GNU/Linux
        vrml_image                      debian.png
```

```
        statusmap_image                 base/debian.gd2
        }
```

A configuração do arquivo **generic-host_nagios2.cfg** especifica um *template* e não um *host* real:

```
define host{
        name                            generic-host    ; The name of this host template.
        notification_enabled            1       ; Host notifications are enabled.
        event_handler_enabled           1       ; Host event handler is enabled.
        flap_detection_enabled          1       ; Flap detection is enabled.
        failure_prediction_enbled       1       ; Failure prediction is enabled.
        process_perf_data               1       ; Process performance data
        retain_status_information       1       ; Retains status information across program restarts
                check_command           check-host-alive
                max_check_attempts      10
                notification_interval   0
                notification_period     24x7
                notification_options    d,u,r
                contact_groups          admins
        register                        0       ; Don't register this definition – Its not a real host, just a template!
}
```

Configuração do arquivo **generic-service_nagios2.cfg** :

```
define service{
        name                            generic-service ; The 'name' of this service template
        active_checks_enabled           1       ; Active service checks are anabled
        passive_checks_enabled          1       ; Passive service checks are enabled / accepted
        paralleze_check                 1       ; Active service checks should be parallelized (disabling this cam lead to major performance problems)
        obsess_over_service             1       ; We should obsess over this service (if necessary)
        chek_freshness                  0       ; Default is to NOT check service 'fresheness'
        notifications_enabled           1       ; Service notifications are enabled
        event_handler_enabled           1       ; Service event handler is enabled
        flap_detection_enabled          1       ; Flap detection is enabled
        failure_prediction_enabled      1       ; Failure prediction is enabled
        process_perf_data               1       ; Process performance data
```

```
        retain_status_information       1       ; Retain status information across program
restarts
        retain_nonstatus_information    1       ; Retain non-status information across
program restarts
                notification_interval    0       ; Only send notifications on status change
by default.
                is_volatile              0
                check_period             24x7
                normal_chek_interval     5
                retry_check_interval     1
                max_check_attempts       4
                notification_period      24x7
                notification_options     w,u,c,r
                contact_groups           admins
        register                        0       ; Don't register this definition – its not a
real service, just a template!
}
```

Configuração do arquivo **hostgroups_nagios2.cfg** :

```
# Some generic hostgroups definitions. A simple wildcard hostgroup

define hostgroup {
        hostgroup_name          all
                alias           All Servers
                members         *
}

# A list of your Debian GNU/Linux servers

define hostgroup {
        hostgroup_name          debian-servers
                alias           Debian GNU/Linux Servers
                members         localhost
}


# A list of your web servers

define hostgroup {
        hostgroup_name http-servers
                alias           HTTP servers
                members         localhost
}
# A list of your ssh-accessible servers

define hostgroup {
```

```
        hostgroup_name          ssh-servers
                alias           SSH servers
                members         localhost
}
```

Configuração do arquivo **localhost_nagios2.cfg** :

```
# A simple configuration file for monitoring de local host. This can serve as an example
for configuring other servers; Custom service specific to this host are  added here, but
services defined in nagios2-common_services.cfg may also apply.

define host{
        use             generic-host        ; Name of host template to use
        host_name       localhost
        alias           localhost
        address         127.0.01
        }

# Define a service to check the disk space of the root partition on the local machine.
# Warning if <20% free, critical if < 10% free space on partition.

define service{
        use                             generic-service         ; Name of service template
to use
        host_name                       localhost
        service_description             Disk Space
        check_command                   check_all_disks!20%!10%
}

# Define a service to check the number of currently logged in users on the local
# machine. Warning if > 20 users, critical if > 50 users.

define service{
        use                             generic-service         : Name of service template
to use
        host_name                       localhost
        service description             Current Users
        check_command                   check_users!20!50
}
```

```
# Define a service to check the number of currently running procs on the local

# machine. Warning if > 250 processes, critical if > 400 processes.

define service{
        use                             generic-service         : Name of service template
to use
        host_name                       localhost
        service description             Total Processes
        check_command                   check_procs!250!400
}


# Define a service to check the load on the local machine.

        define service{
        use                             generic-service         : Name of service template to use

        host_name                       localhost
        service description             Current Load
        check_command                   chek_load!5.0!4.0!3.0!10.0!6.0!4.0
}
```

Configuração do arquivo **services_nagios2.cfg** :

```
# check that web services are running

define service {
        hostgroup_name                  http-servers
        service_description             HTTP
        check_command                   check_http
        use                             generic-service
        notification_interval           0       ; set > 0 if you want to be renotified
}

# check that ssh services are running

define service {
        hostgroup_name                  ssh-servers
        service_description             SSH
        check_command                   check_ssh
        use                             generic-service
        notification_interval           0       ; set > 0 if you want to be renotified
}
```

Configuração do arquivo **timeperiods_nagios2.cfg** :

```
# This definers a timeperiod where all times are valid for check, notifications, etc.

# The classic “24x7” support nightmare.

define timeperiod{
        timeperiod_name         24x7
        alias                   27 hours A Day, 7 Days A Week
        sunday                  00:00-24:00
        monday                  00:00-24:00
        tuesday                 00:00-24:00
        wednesday               00:00-24:00
        thursday                00:00-24:00
        friday                  00:00-24:00
        saturday                00:00-24:00
}


# Here is a slightly friendlier period during work hours

define timeperiod{
        timeperiod_name         workhours
        alias                   Standard Work Hours
        monday                  09:00-17:00
        tuesday                 09:00-17:00
        wednesday               09:00-17:00
        thursday                09:00-17:00
        friday                  09:00-17:00
}


# The complement for workhours

define timeperiod{
        timeperiod_name         nonworkhours
        alias                   Non-Work Hours
        sunday                  00:00-24:00
        monday                  00:00-09:00,17:00-24:00
        tuesday                 00:00-09:00,17:00-24:00
        wednesday               00:00-09:00,17:00-24:00
        thursday                00:00-09:00,17:00-24:00
        friday                  00:00-09:00,17:00-24:00
        saturday                00:00-24:00
}


# This one is a favorite: never :)
```

```
define timeperiod{
        timeperiod_name         never
        alias                   Never
}

# End o file
```

Depois de efetuadas as configurações dos arquivos do Nagios, tomando-se por base as descrições acima, apresenta-se a estrutura da rede do caso estudado.

Na Figura 6 o *nó* central da rede, SSR8 encontra-se no Prédio A; o *nó* SW-PALÁCIO encontra-se no Prédio B e o *nó* SW-VERDINHO, localiza-se no Prédio C.

Na Figura 7 destacam-se os serviços básicos nos servidores conectados ao *nó* SW-SERVERS, conforme se pode ver no Mapa da Rede.

Na Figura 8 destacam-se o *status* de alguns *hosts* os quais estão conectados ao *nó* central, o SSR8.

Na Figura 9 destacam-se os *Services Actively Checked* e os *Hosts Actively Checked* nos intervalos especificados na figura e os percentuais correspondentes.

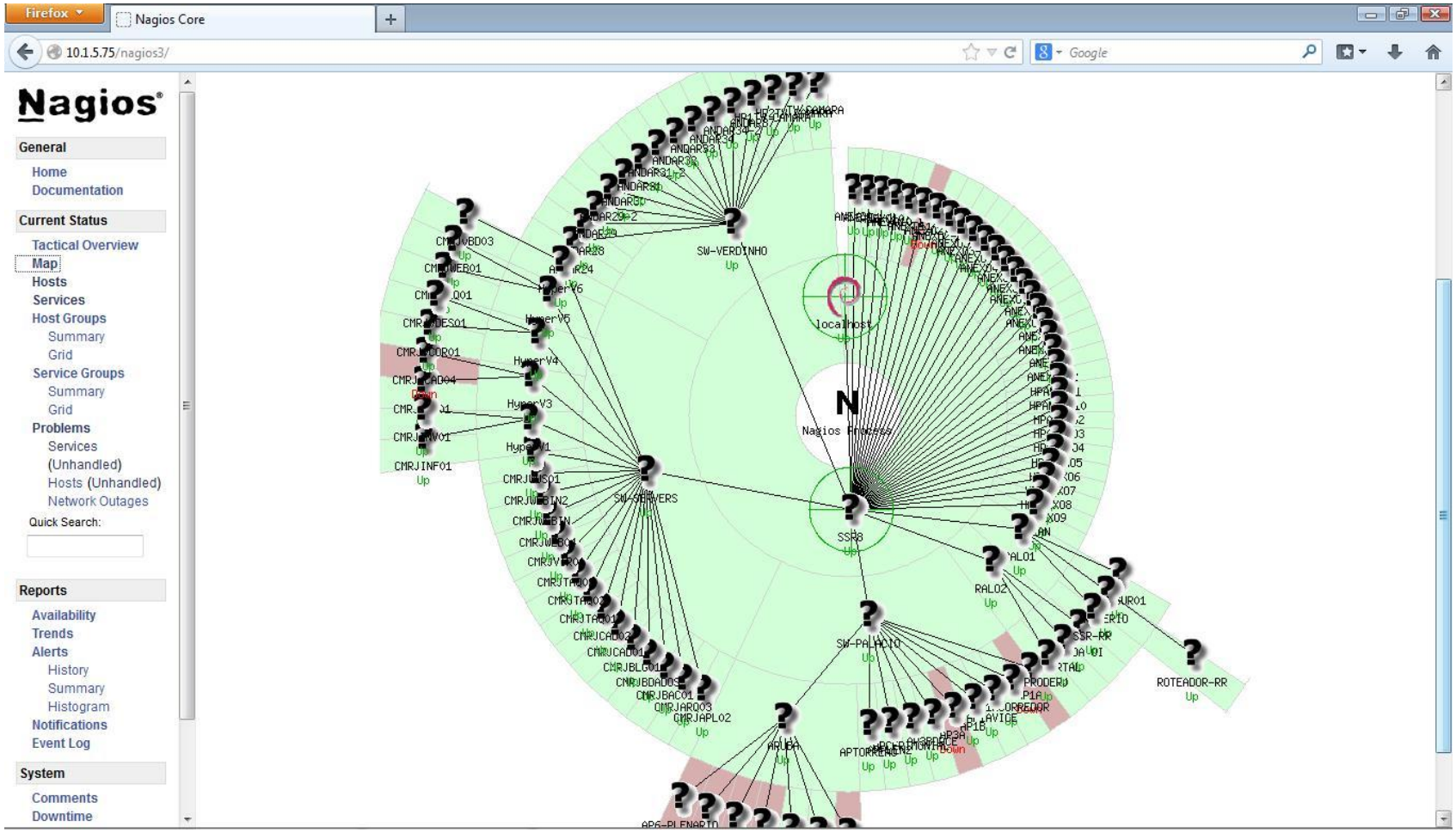

Figura 6 – O Mapa da Rede

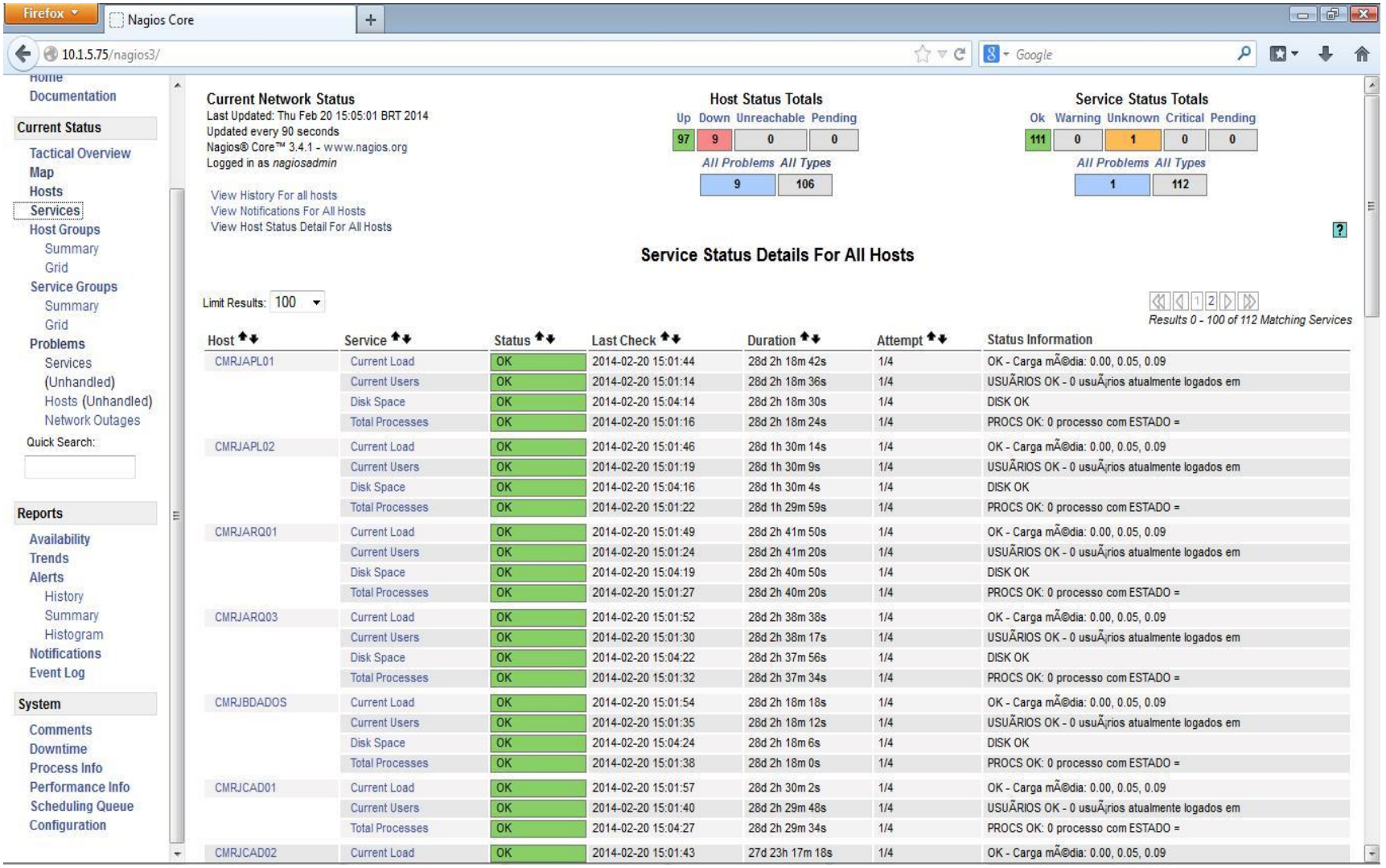

Figura 7 – Os serviços monitorados.

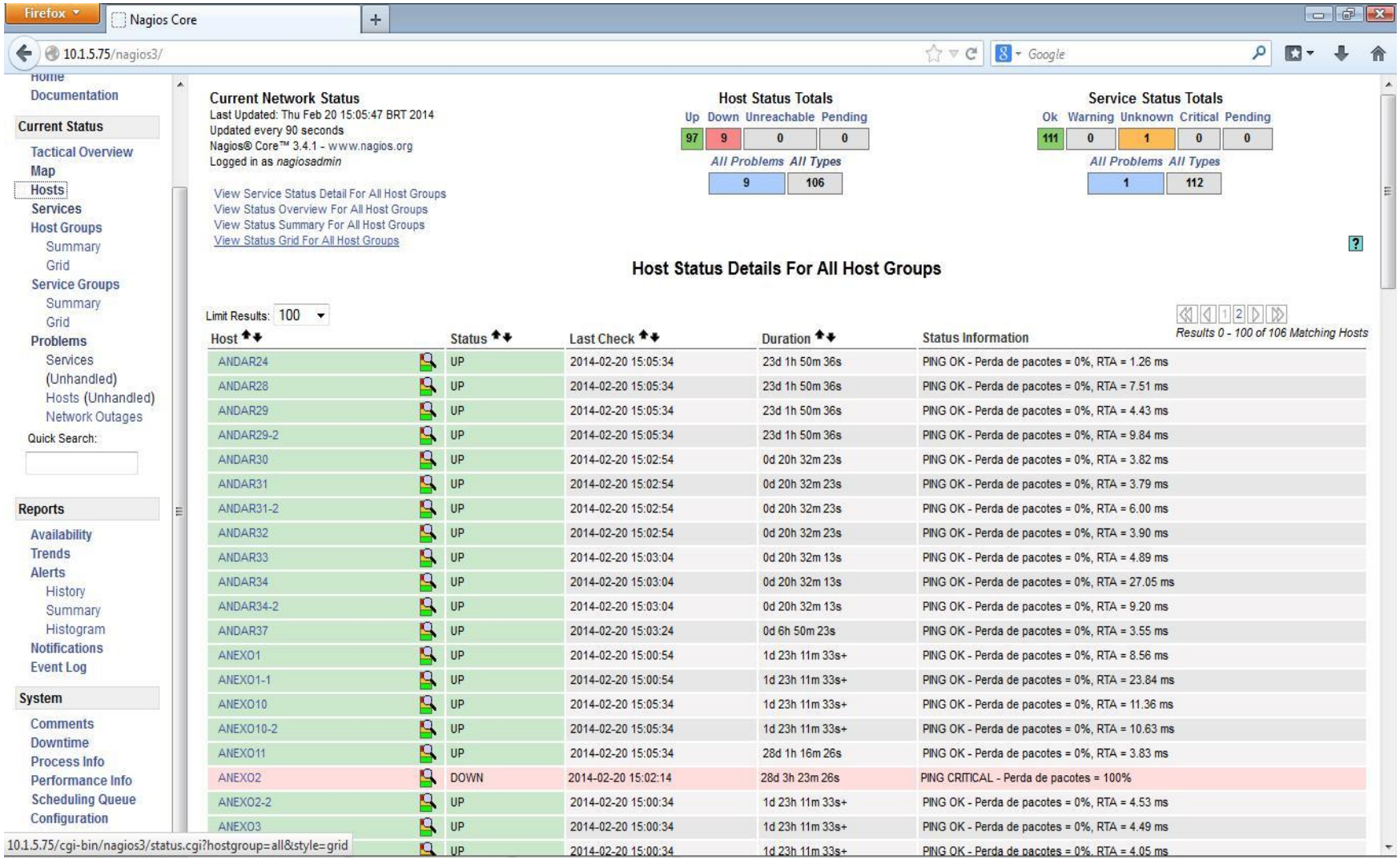

Figura 8 – Relação de hosts monitorados.

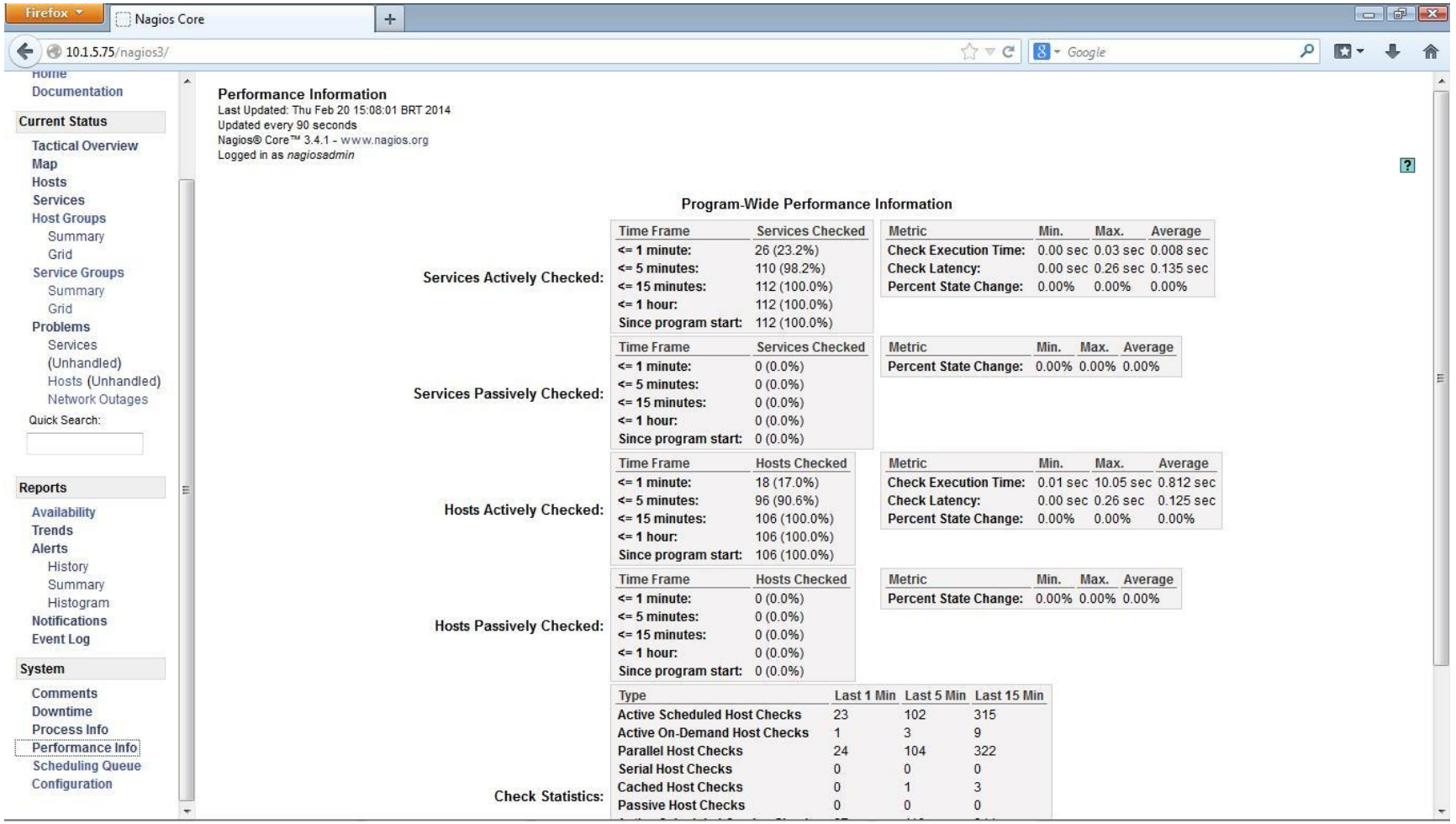

Figura 9 – A performance da Rede

6 CONCLUSÃO

O gerenciamento de redes é cada vez mais importante para as organizações, à medida que utilizam suas redes corporativas como recursos indispensáveis para o seu funcionamento administrativo e operacional.

A larga utilização da Internet fez com que houvesse muitos investimentos nas tecnologias de redes, aumentando em várias vezes o volume e a velocidade de transferência de dados. O uso dessas tecnologias abrangeu as redes corporativas, o que gerou problemas para o gerenciamento, em virtude de seu protocolo mais utilizado – o SNMP – não ter acompanhado essa evolução.

O aumento de informações disponíveis na Internet fez com que o número de *hackers* se elevasse consideravelmente, pois qualquer usuário poderia ter acesso a *softwares* como os analisadores de tráfego, gerando mais problemas para os gerentes de redes, pelo fato do SNMP utilizar uma *string* de texto para trafegar sua senha de autenticação. No caso de captura dessa senha, o *hacker* poderia até desligar equipamentos monitorados na rede, podendo gerar enormes prejuízos para as corporações.

A evolução do protocolo para a sua segunda versão foi problemática, principalmente na questão da segurança, uma vez que divergências entre seus desenvolvedores fizeram com que muitas características fossem abandonadas.

Entretanto, a terceira versão do SNMP agregou as melhores características da segunda versão e incorporou um *framework* para segurança de mensagens e controle de acesso, medidas estas que fizeram o SNMPv3 se tornar uma das melhores opções para o gerenciamento de redes da atualidade.

Com relação à implementação deste estudo de caso na instituição se pode concluir que hoje todos os dispositivos são monitorados na rede; situação não existente anteriormente.

As dificuldades de identificar eventuais falhas em *switches*, *concentrador da rede* e servidores, principalmente; eram problemas que causavam transtornos à instituição e aos administradores de TI.

Após a conclusão desse trabalho pode-se afirmar que o Mapa da Rede, o monitoramento dos hosts, dos diferentes serviços e o desempenho da Rede, permitem um nível de segurança que anteriormente não era visível para os administradores de TI, desta forma constituindo-se numa ferramenta de informações extremamente valiosa na manutenção e desenvolvimento de futuros projetos.

REFERÊNCIAS

CASTRO, Oscar. Gerência Estratégica SNMP. NCE-UFRJ, 2012.

COMER, Douglas. Interligação em rede com TCP/IP, Volume I. Editora Campos: Rio de Janeiro, 1998.

COMER, Douglas. Interligação em redes com TCP/IP, Volume II. Editora Campos: Rio de Janeiro, 1999.

DAVIN, J. et al. A Simple Gateway Protocol, RFC 1028. 1987.

ENGER, R. & REYNOLDS, J. FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices, RFC 1470. 1993;

HEGERING, Heinz-Gerd; ABECK, Sebastian et NEUMAR, Bernhard. Integrated Management of Networked Systems: concepts, architectures, and their operational application. Morgan Kauffmann Publishers: São Francisco, 1998.

GASTALD, Ethan. Nagios Core 3.x Documentation. 2012. Site http://www.nagios.org.

MAURO, D. & SCHIMIDT, K. SNMP Essencial. Editora Campos: Rio de Janeiro, 2001.

REYNOLDS, J. & POSTEL, J. Assigned Numbers, RFC 1060. 1990.

SIMIER, Pierrick, www.snmplink.org

STALLINGS, Willian. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Addison Wesley, 1999.